(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 March 2000 (09.03.2000)

PCT

(10) International Publication Number
WO 00/13099 A1

(51) International Patent Classification⁶: G06F 15/16

(21) International Application Number: PCT/US99/19074

(22) International Filing Date: 19 August 1999 (19.08.1999)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/143,726 31 August 1998 (31.08.1998) US

(71) Applicant: CUBUS CORPORATION [US/US]; Suite 300, 651 Brannan Street, San Francisco, CA 94107 (US).

(72) Inventors: STIEFEL, James, G.; 942 Fell Street, San Francisco, CA 94117 (US). KHAN, Azhar, H.; 5814 Nottingham Drive, Oakland, CA 94611-2165 (US). COLLIER, Zaitrarrio, T.; 430 Merritt Avenue, #301, Oakland, CA 94610 (US). LEE, Benjamin, J.; 128 Saddler Avenue, Colonial Heights, VA 23834 (US).

(74) Agents: BROOKS, Kenneth, C. et al.; Townsend and Townsend and Crew LLP, Two Embarcadero Center, 8th floor, San Francisco, CA 94111-3834 (US).

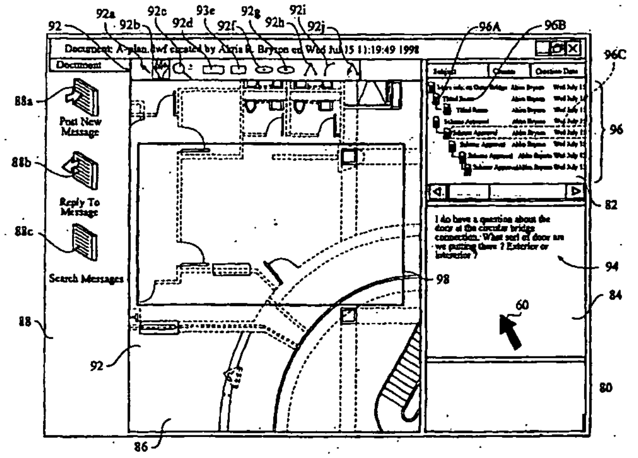
(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: COMPUTER PRODUCT FOR NETWORKING A DOCUMENT DEVELOPMENT SYSTEM USING MESSAGE HEADERS ASSOCIATED WITH MESSAGE FILES



I do have a question about the door at the circular bridge connection. What sort of door are we putting there ? Exterior or interior ?

(57) **Abstract:** A computer product that facilitates networking a document development system which bifurcates data corresponding to a document (92) under development to minimize a quantity of data transmitted to a client terminal (12), while providing to the client terminal (12) sufficient data to display a history of the development of the document (92). The computer product includes storing on a server (10) a plurality of message files (94), and creating a plurality of message headers (96), each of which corresponds to one of the plurality of message files (94), and storing on the server (10) the document (92) under development, which has visually perceivable information associated therewith. A subportion (98) of the visually perceivable information is defined by a plurality of view parameters, with a subset of the plurality of message files (94) having differing view parameters associated therewith. A subgroup of the message headers (96) are displayed on the client terminal (12), each of which recites a summary of content of one of the plurality of message files (94) corresponding to the message headers (96) of the subgroup.

(48) Date of publication of this corrected version:
13 December 2001

(15) Information about Correction:
see PCT Gazette No. 50/2001 of 13 December 2001, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

COMPUTER PRODUCT FOR NETWORKING A DOCUMENT DEVELOPMENT SYSTEM USING MESSAGE HEADERS ASSOCIATED WITH MESSAGE FILES

BACKGROUND OF THE INVENTION

The present invention relates to programmable computer systems. More particularly, the present invention is directed to an apparatus suited for dynamic design collaboration over a data network.

Data networks, such as the "Internet" have facilitated collaborative efforts of several individuals remotely disposed from one another. The Internet typically includes a plurality of users employing client computers communicating with a remote server computer to transfer information therebetween. To facilitate the transfer, the client computers have a "web" browser that provides graphical user interface (GUI)-based communication with a "web page" obtained from a server. One popular collection of servers uses a standardized Hypertext Transfer Protocol (HTTP) to provide information and is known as the "World Wide Web." The information is typically presented as web pages written as text with standardized formatting and control symbols known as Hypertext Mark-up Language (HTML). HTML provides basic document formatting and allows a server to specify "links" to other servers and files. Use of an HTML-compliant browser involves specification of a link via a Uniform Resource Locator (URL). Upon such specification, the user's client computer makes a TCP/IP request to the server identified in the link and receives an HTML file that is interpreted by the browser so that a electronic HTML document made up of one or more web pages may be displayed on the client's computer.

A drawback with the Internet data transmission, however, is that it is difficult to provide real time document development involving either large numbers of iterations of a document or contributions of several individuals to a documents development. Specifically, documents being developed are large which can substantially increase data transfer time between the client computer and the server computer that can be frustrating to an end user.

What is needed, therefore, is an integrated document development method that facilitates dynamic design collaboration by multiple developers over a data network such as the Internet by minimizing data transfer time.

SUMMARY OF THE INVENTION

The present invention provides a computer product for networking documents undergoing development that bifurcates data corresponding to a document under development to minimize a quantity of data transmitted to a client terminal while providing, to the client terminal, sufficient data to display a history of the development of the document. To that end, the computer product operates on a network having a server and a client terminal. The client terminal includes a display. The server stores the document under development and a plurality of message files, as well as a plurality of message headers. Each of the plurality of message headers corresponds to one of the plurality of message files. The document under development has visually perceivable information associated therewith, a subportion of which is defined by a plurality of view parameters. A subset of the plurality of message files have differing view parameters associated therewith. A subgroup of the message headers are displayed on the client terminal, each of which recites a summary of content of one of the plurality of message files corresponding to the message headers of the subgroup. The client terminal includes memory and further including a step of placing the document file and one of the message files of the subset in the memory, with the viewable information associated with the document file being related to the view parameters associated with message file of the subset present in the memory.

The document file and the message file may be any type of computer readable file. Examples of the documents file includes vector-based files, character based files and raster files. The message file may consist of any of the aforementioned files, as well as an audio file or video streams. In a preferred embodiment, the method and system is employed over a data network, such as a local area network, a wide-area data network, e.g., the Internet and the like.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is simplified plan view of a computer network in which the present invention is implemented;

Fig. 2 is a block diagram of a client terminal shown above in Fig. 2;

Fig. 3 is a plan view of a web page having a visual representation of a hypertext link employed to access the present invention;

Fig. 4 is a plan view of a second web page having a visual representation of a hypertext link employed to access the present invention;

Fig. 5 is a plan view of a web page employed to allow users to either log-in or register to gain access to the present invention;

Fig. 6 is a plan view of a dialog box employed to allow a new user to register to gain access to the present invention;

Fig. 7 is a plan view of a window upon which a list of projects having documents under development are recited in accordance with the present invention;

Fig. 8 is a plan view of a window in which a document file, a message file and a plurality of message headers are concurrently displayed;

Fig. 9 is a plan view of a window employed to post a new message file;

Fig. 10 is a plan view of the window shown in Fig. 8 with the same document file being displayed along with a different message file and in accordance with a set of view parameters which differ from those associated with the document file shown in Fig. 8;

Fig. 11 is a simplified plan view of the major classes of objects associated with computer code employed to practice the present invention;

Fig. 12 is a flow diagram showing steps for storing a message file in accordance with the present invention; and

Fig. 13 is a flow diagram showing steps for retrieving a message file in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Fig. 1, a large area network is shown, such as the Internet, which includes a plurality of networked "servers" 10 that are accessible by "client terminals" 12. Communication between the servers 10 and the client terminals 12 typically occurs over a publicly accessible network, such as a public switched telephone network over ASDL telephone lines or large bandwidth trunks, such as T1 or OC3 service. The client terminals 12 access the various servers 10 through an Internet service provider, e.g., America On-Line, Prodigy, CompuServe and the like, by executing application specific software, commonly referred to as a "browser", on a computer 14, shown more clearly in Fig. 2.

Referring to Fig. 2, the computer 14 includes a system unit 20 having one or more system buses 22 placing various components of the system in data communication. For example, a microprocessor 24 is placed in data communication with both a read only memory (ROM) 26 and random access memory (RAM) 28 via the system bus 22. The ROM 26 contains among other code, the Basic Input-Output system (BIOS) which controls basic hardware operation such as the interaction with peripheral components such as disk drives 30 and 32, as well as the keyboard 34. The RAM 28 is the main memory into which the operating system and application programs are loaded and affords at least 32 megabytes of memory space. The memory management chip 36 is in data communication with the system bus 22 to control direct memory access (DMA) operations. DMA operations include passing data between the RAM 28 and the hard disk drive 30 and the floppy disk drive 32. Also in data communication with the system bus 22 are various I/O controllers: a keyboard controller 38, a mouse controller 40, a video controller 42, and an audio controller 44. The keyboard controller 38 provides a hardware interface for the keyboard 36, the mouse controller 40 provides the hardware interface for a mouse 46, or other point and click device, and the video controller 40 provides a hardware interface for a display 48. A modem 50 enables data communication over the network facilitating data transmission speeds of at least 28.8 kilobytes per second. The operating system 52 of the computer 14 may be DOS, WINDOWS 3.x, WINDOWS 95, OS/2, or other known operating system. Preferably, the operating system is WINDOWS NT 4.0 or WINDOWS 95. The RAM 28 also supports a number of Internet access tools, including, for example, an HTTP-compliant web browser having a JavaScript interpreter, such as Netscape Navigator 3.0, Microsoft Explorer 3.0 and the like.

Referring to both Figs. 1 and 2, the integrated development system and method is accessed over the Internet. The browser 54 employs a TCP/IP connection to pass a request to a server 10 running an HTTP "service" (under the WINDOWS operating system) or a "daemon" under the UNIX operating system. The request is typically achieved by contacting an HTTP server 10 at the following address "http://www.cubus.net," employing a protocol that can be used to communicate between the server 10 and the client terminal 12. The HTTP server 10 then responds to the protocol, typically by sending a "web page" formatted as an HTML file. The browser 54 interprets the HTML file and may form a visual representation of the same using local resources, e.g., fonts and colors.

Referring to Figs. 2 and 3, the "web page" 56 displayed at the aforementioned address includes, inter alia, a plurality of hypertext links, shown generally as 58a-g. Employing the mouse 46, a cursor 60 may be placed proximate to hypertext link 58d, entitled ReviewIt$_{TM}$ and a cursor event is effectuated which connects to web page 62, shown in Fig. 4. Web page 62 includes a plurality of hypertext links 64a-i. Hypertext link 64a allows downloading of the requisite ReviewIt software to interact with the server to practice the present invention. Hypertext link 64b is activated to practice the present invention once the requisite software is loaded onto the client terminal.

Log-in is achieved by activating hypertext link 64b which causes the ReviewIt software to run and display a log-in screen 66, shown in Fig. 5. The log-in screen includes a dialog box 68 centered within a project view window 70. The dialog box 68 includes a plurality of data entry fields 68a and 68b. Also included thereon is a virtual button 68c which allows submission of the data once entered in fields 68a and 68b, as well as virtual button 68d which allows registering to use the present invention. One of the data entry fields 68a is to receive a user's e-mail address. The e-mail address is employed as a ReviewIt$_{TM}$ log-in identifier, discussed more fully below. To restrict access, a password data entry field is provided 68b in which any series of number or letters, up to 256 characters, may be entered. As is standard with most password security features, password data is never recited in the field 68b.

If a user desires to register to use the present invention, virtual button 68d is activated and dialog box 71 is displayed so as to be in superimposition with dialog box 68, shown in Fig. 6. Dialog box 71 includes a plurality of data entry fields 71a-p. Also included thereon are virtual buttons 71q and 71r which allows submission of the data, once data has been entered in fields 71a-p, and cancellation of the registration, respectively. Once the registration data has been successfully submitted, dialog box 71 is removed, exposing dialog box 68, once again. Log-in is then achieved as discussed above.

Upon acceptance of the log-in data, a TCP/IP connection to an additional server, which is the ReviewIt$_{TM}$ server 10a, is achieved and a project view window 70 is completely displayed, shown in Fig. 7. The project view window 70 is segmented into a plurality of regions: the menu bar 72; the project list 74; and the project file directory 76. Each of the regions may be adjusted to occupy areas of differing sizes on the display 12a. Typically, the project view window 70 is sized to fit the browser 54 window, with the project view menu bar 72 located on the left side of the display 12a and the project list 74 disposed atop of the project file directory 76 on the right-hand side.

The project list region 74 includes four columns of information identified as the following: project name; description, creator and creation date. The widths of the columns may be adjusted using the mouse 46 by initiating a cursor event proximate to the edge of the window as is well known in the computer graphics art. In addition, the files recited in the Project list may be sorted by initiating a cursor event proximate to one of the column headings, e.g., sorted by project listing by name, creator, etc. The project name recites the names of each project, and if the files associated therewith have been modified, this is indicated by having the project name have an appearance which differs from non-modified files. For example, a project with modified files may be displayed in a different color from projects with non-modified files. The description column recites a brief description of the purpose of the project name positioned adjacent thereto. The creator column identifies individuals who were primarily responsible for the creation of the project having a name in the column, and the creation date identifies the date that the creators originated the project, e.g., date and time. Usually, the information recited in the columns does not change for a project, once created.

The menu bar 72 contains a plurality of virtual buttons 78a-78f. Button 78a, for example, facilitates adding projects to the project list 74 and contains folders and documents, which may be organized by a creator and selectively shared with additional users. Similarly, virtual buttons 78b and 78c facilitate adding folders and documents, respectively. Documents are stored within folders and prioritized according to whatever constructs pertinent to a user. Documents are typically uploaded to the server. Virtual button 78d facilitates modifying user information, such as passwords and log-in identifier data. Virtual button 78e facilitates modifying preferences, such as modify access privileges of users for differing projects. Finally, button 78f allows logging into or out of the ReviewIt$_{TM}$ server 10a.

The project file region 76 recites the contents of each project listed in the project list region 74 in a threaded type of file hierarchy system. Each file and folder listing is divided into four columns having the headings discussed above with respect to the document list directory.

To access a particular file, a user highlights the requisite project with the cursor 60 by placing the same proximate to one row of text in the project list region 72. The selected column is then highlight and a cursor event is initiated by twice activating a button (not shown) on the mouse 46. This provides a list of folders and files associated with the aforementioned project, in the project file region 76. Specifically, folders have [+/-] signs adjacent thereto are indicated as having files associated therewith. The [+] indicates that the files are not displayed and the [-] sign indicates that the files are visible. To conceal visible files, a cursor event is initiated proximate to the [-] sign. Thereafter, a file is displayed by highlighting the same and initiating a cursor event. This produces a document view window 80, shown in Fig. 8.

Referring to Both Figs. 2 and 8, the document view window 80 provides an integrated view of a document under development. To that end, the document view window 80 is segmented into a plurality of regions 82, 84, 86, 88 and 90 on the display 48. Displayed in one of the regions 86 is a document file 92. Displayed in a second of the regions 84, is a message file 94 and a plurality of message headers 96 are displayed in a third region 82. Displayed in a fourth region 90 is a tool palette 91. Displayed a fifth region 88 is a plurality of virtual buttons 88a, 88b and 88c.

The document file 92 may be any file format, e.g., a vector-based file such as Drawing Web Format (DWF), Autodesk AutoCAD (DWG), Drawing Exchange Format (DXF), Simple Vector Format (SVF), Bentley Microstation (DGN, CEL), Adobe Illustrator (AI), Corel Draw (CMX), Microsoft Powerpoint (PPT), HPGL and HPGL/2. Additionally, the document file 92 may have a raster file format, such as Windows Bitmap (BMP), Compuserve GIF (GIF), Tagged Image File Format (TIFF), Joint Photographic Experts Group (JPEG or JPG), Device Independent Bitmap (DIB), Targa (TGA) or PCX, as well as a hybrid Formats, such as Hypertext Markup Language (HTML), Adobe Portable Document Format (PDF), Windows Metafile (WMF), Postscript (PS, EPS). If desired, document file may have a text format, such as, ASCII text (TXT), Rich Text Format (RTF), Microsoft Word (DOC) and the like. Similarly, the content of the message file 94 may include any of the aforementioned file formats, as well as a video stream, such as Motion Picture Experts Group (MPEG) Microsoft Video for Windows (AVI) Quicktime (MOV), an audio file or combination thereof. An advantage of this arrangement of information on the display 48 is that it provides an integrated document development system which allows simultaneously displaying a document being developed, detailed changes made thereto, as well as a brief history of the changes made during the development process. Further, an unedited version of the document being developed may be accessed at any stage during the development. The document file 92 includes data defining the document being developed, and the content of the message file 94 typically includes data corresponding to at least a portion of the viewable information associated with the document file 92. For example, a subportion of the viewable information of document file 92 is enclosed by a rectangle employing the tool palette 91, defining an annotation 98 which is discussed more fully below. The content of the message file 94 is related to the annotation 98. In the present example, the message file 94 recites "I do have a question about the door at the circular bridge connects . . . ." The document file 92 shows a visual depiction of information associated with the content of the message file 94. In the present example, the document file 92 has viewable information corresponding to building plans, i.e. "blue print", and the area, identified by the document file 92, where a door would be located. In this fashion, both a visual presentation and a detailed textual description of a common idea is expressed.

To that end, each message header 96 provides a summary of the content of a message file 94. Typically, a plurality of message headers 96 are provided, each corresponding to a different message file 94 and having an icon 96a and a line of text 96b disposed adjacent thereto. With the plurality of message headers 96 being displayed in region 82, a brief history of the message files 94 associated with the document file 92 is provided. The message header 96 corresponding to the message file 94 being displayed has a background 96c which optically contrasts with the background associated with the remaining message headers 96, i.e., the message header 96 is "highlighted". As seen in Fig. 8, the highlighted message header 96 recites "Scheme Approval". The message file 94 associated therewith recites "I do have a question about the door at the circular bridge connects . . . ." With this arrangement, the message file 94 and the message header 96 both provide a description of a subportion of the viewable information, typically the subportion corresponding to the annotation 98.

The tool palette 91 has one or more tool-defining regions, shown as 92a-92j and is employed to create the annotation 98. Each of the tool-defining region 92a-92i specifies a predetermined operation that modifies a subportion of the viewable information of the document file 92, defining an annotation 98. An example of an annotation 98 is shown as an oval, however, the annotation 98 can be of any shape desired and may include alphanumeric characters, depending upon the tool-defining regions of the tool palette 91, as well as the file format of the document file 92. For purposes of the discussion, the document file 92 will be discussed as being a vector-based file commonly associated with computer aided design programs and systems. Thus, part of the viewable information is defined by view parameters, such as magnification, rotation, layer, window coordinates. As a result, the tool palette 91 provides tool-defining regions which support, inter alia, operations associated with aforementioned view parameters. Examples of such tool defining regions are a pointer tool 92a employed to select areas of the document file 92 to be modified, such as by deletion.

A hand tool 92b may be employed to pan the viewable information associated with the document file 92 along any direction in the plane of the region 86. A zoom tool 92c may be employed to increase or decrease the resolution of the viewable information. A box tool 92d·facilitates surrounding an area of the viewable information with a rectangle or other polyhedron, and a filled box tool 92e makes the viewable information within the area opaque. Other shapes may be provided to achieve the same results by providing an oval tool 92f and a filled oval tool 92g. Other shape tools may be user defined with an irregular object tool 92h. An arrow tool 92i allows pointing to a particular section of the viewable information, and an arc tool 92j is employed to create arcuate lines.

Referring to both Figs. 8 and 9, to post new message files, or reply to an existing message file 94, the virtual buttons 88a and 88b are provided. Specifically, to post a new message file, virtual button 88a is activated by execution of a cursor event proximate thereto. This produces a message file, in a window 95, having no content associated therewith, and the document file 92 corresponding thereto is in an original state. In this example, an original state is defined as being at minimum magnification, with no annotation 98 present therein. Thus, viewable information corresponding to a project is displayed in region 86. To produce the information associated with Fig. 8, a user would activate tool-region 92c and magnify the document file 92 to the desired resolution providing the detail necessary to clearly provide a visual depiction of the message file 94, generating modified view parameters that are associated with the viewable information.

Locating a desired position in the document file 92 may be achieved by employing the hand tool, associated with tool-defining region 92b. To precisely delineate the area of the viewable information of the document file 92 being described, tool-region 92e is activated and the cursor 60 is placed on the region shown therein, producing annotation 98. After the portion of the viewable information of document file 92 has been located, the content of the message file 94 corresponding thereto is entered by placing the cursor 60 over the window 95 and initiating a cursor event. Text is entered using a standard QWERTY keyboard.

Additionally, a graphics file or an audio file may be entered into the message file 94, such as graphics file 89, shown in Fig. 10. After the desired content is present in the message file 94, the message is sent to the ReviewIt$_{TM}$ server 10a via initiating a cursor event occur proximate to virtual button 97. The message is saved on the ReviewIt$_{TM}$ server 10a with the view parameters. As the message header 96 corresponds to a newly created message file 94, the icon 96a of the same is located to the left side of the region 82. This produces a message header 96 in region 82 which corresponds to the newly created message file 94 in region 82.

Upon retrieval, the modified viewable information associated with the document file 92 is displayed as it appeared in region 86 when the message file 94 was sent, i.e, the same magnification and identical annotation 98 in the area of the document file 92 displayed. This is referred to as context preservation and is discussed more fully below. However, all of the viewable information associated with the document file 92 is accessible. This allows a user to manipulate the viewable information associated therewith so as to display a completely different area of the document file 92 at a completely different magnification. Upon exiting and retrieving the same message file 94, the modified view parameters are restored along with the modified viewable information.

Accessing a particular message file 94 may be achieved employing either the message headers 96 or annotations 98 on a document file 92. For example, highlighting a particular message header 96 and initiating a cursor event displays a message file 94 with content summarized by the highlighted message header 96. Were a document file 92 displayed to have multiple annotations 98 present thereon, the cursor 60 may be positioned to highlight the desired annotation 98. Thereafter a cursor event would display the message file 94 associated with the annotation 98, referred to as graphical indexing. Moreover, a boolean search technique may be employed by highlighting virtual button 88c and initiating a cursor event. Thereafter a dialog window (not shown) would appear in the window 80 and provide an area to enter search terms, as is well known in the prior art.

Referring to Figs. 8 and 9, to reply to an existing message, virtual button 88b is activated by initiating a cursor event proximate thereto. Thereafter, analogous steps are taken to post a reply as is necessitated to post a new message, except that the message header 96 corresponding to the reply message is positioned below the selected message to which a reply is being posted, offset from the left side of the region 82. In this fashion, not only do the plurality of message headers 96 provide a historical summary of the message files corresponding thereto, but they also describe the sequence of communications between users posting the same. The spatial position of each message header 96 is dependent upon a position of the communication, corresponding thereto, amongst the sequence communications. In this manner, the ReviewIt$_{TM}$ server 10a provides an entire history of the development of a document in a centralized location.

By centralizing the location of the document files, as discussed above, various techniques may be employed to further control and manage document development. For example, a creator of a project may be sent an e-mail each time a project is accessed by a user. Various information may be supplied to the creator, including the project and files accessed, the duration of the access, any message files 94 posted and the modifications, if any, recited therein. Further, the cost implication of modifications to a project may be modeled. In addition, authenticity and authorization of modifications to a project may be controlled by implementation of electronic signature techniques. This could be implemented to reduce cost overruns, particularly useful in public sector civil engineering contracts.

Referring again to Fig. 1 an important aspect of the present invention is to provide high-speed data communication between the client terminals 12 and the ReviewIt$_{TM}$ server 10a while providing the full flexibility of the integrated system. To that end, the ReviewIt$_{TM}$ server 10a bifurcates the data transmitted to a client terminal 12, concerning a document under development, into single and multiple server call data. The single server call data is transmitted to the client terminals 12 as soon as a project is opened. The multiple server call data is transmitted after an additional server call has been transmitted by the client terminals 12 after a project has been opened. Examples of single server call data are the message headers 96 and the annotations 98. Examples of multiple server call data is the message content and the view parameters. Upon initializing a server call, a client terminal 12 immediately receives the message header 96 and the annotations 98 associated with a document file 92. This provides the information necessary for a user to quickly view the history of message files 94 concerning a document file and determine which message is desired. Upon finding the requisite message header 96, a user initiates a cursor event, thereby taking advantage of message context preservation, as discussed above. Specifically, the contents of the message file 94 are displayed, as well as, the document file 92, with the document file 92 being displayed in accordance with the view parameters previously modified. Alternatively, a message file 94 could be displayed employing the graphical indexing technique discussed above. In this fashion, a user would place the cursor 60 on an annotation 98 desired and initiate a cursor event. This generates a server call over the network, retrieving the content information of the message file 94 associated therewith, along with the view parameters, if any.

Referring to Fig. 11, to achieve the aforementioned data communication, it is preferred that the code employed to implement the present invention is object based, such as C++. The objects are divided into a plurality of classes. The classes include a message explorer class 202, a document viewer class 204, a node view class 206, a coordinator core class 208, a document class 210, and a message node class 212. Each of the objects in a class includes data and methods. The data and methods concerning the content of the message files 94 and the view parameters of the document file 92 associated therewith are encapsulated in objects of the message node class 212. Data and methods concerning the document file 92 are encapsulated in objects of the document class 210.

Referring to both Figs. 11 and 12, when creating a message at step 300, an object in the Message Node class 212 is generated, calling a member post method entitled CoreMessageNode::post(). This calls a member method of the Coordinator core class 208 entitled CoordinatorCore::post_message(), which retrieves annotation data, document file data and the view parameters, at step 302. The aforementioned information is retrieved from the Document Viewer class of objects employing the member method of that class entitled the DocumentViewerCore::get_annotation(). The DocumentViewerCore::get_annotation() method retrieves data members of the Document class 210. Specifically, the DocumentViewerCore::get_annotation() method retrieves the serialized annotations employing the serialize_annotation_data() method, as well as the view parameters employing the CubusDocument::get_document_specific_data() method. The aforementioned data members are then associated with the MessageNode class 212, at step 304, allowing the same to be immediately available for subsequent requests. Thereafter, at step 306, the CoordinatorCore::post_message() method continues processing the data members associated with the Message Node class 212 which are then transmitted to the server 10 for long term storage and subsequent retrieval.

Referring to both Figs. 11 and 13, to retrieve a particular message, a user selects a message, at step 400, in the threaded message view by highlighting the same as discussed above. Control passes from the client terminal 12 to the ReviewIt$_{TM}$ server 10a by implementing the SftTreeNodeView::on_selchange() method associated with the objects of the Node View class 206. Employing the get_node() method member of the NodeView class of objects, the data members associated with the message file 94 are retrieved, at step 402. The data members of an object generated by the message being retrieved in the Message Node class 212 contains all the information to that particular message. This information includes, inter alia, view parameters and message content data. The resulting Message Node's select() method is then called.

The Coordinator Class' select_message() is called, passing the previously retrieved Message Node class 212 data members to determine if the desired message is already cached in the client terminal 12, at step 404. This is achieved by testing the values of the cache, in this case, specified addresses in RAM 98, to determine whether a null is present. If the values are not null, then the data is retrieved therefrom at step 406. If a null is present, a request is sent to the server to retrieve the desired information via the request_message() method of the Coordinator class 208, at step 408. The data is then transmitted to the client terminal 12 employing the process_mrsp() method of the Coordinator class 208, at step 410.

The message is implemented in the select_message() method of the Document class. This method retrieves the view parameters employing the get_field() method associated with the Message Node class, which is then passed to the current document file 92 via the document's set_document_specific_info() method associated with the Document class 210, at step 412.

The Message Node class 212 data members are then passed to the Message Explorer class in the update_preview() method. The Message Explorer object get the message content from the data members associated with the Message Node class employing the node's get_field() method. Thereafter, the resulting buffer is passed to the set_formatted_text() method to display it, at step 414. The end result is that the document viewer changes its display to restore the view to the original author was looking at when the message was created.

Although the foregoing has been discussed with respect to a vector-list file, is should be understood that the document file 92 could be a text based file, such as a common wordprocessor file. In this fashion, view parameters associated therewith could include character attributes, e.g., italics, bold, underline and the like, as well as text and graphic boxes.

Furthermore, the above embodiment illustrates use of the present invention in a CAD environment. It should be understood that the embodiments of the present invention can also be applied to other situations in which collaboration is required. Examples of other situations in which the present invention may be applied include health care diagnoses and treatment, advertisement layout and design, movie production and the like. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the appended claims along with their full scope of equivalents.

WHAT IS CLAIMED IS:

1. A computer product of the type comprising a computer readable medium that contains a program to control a network having a server and a client terminal having a display and a memory associated therewith, said computer product comprising:

computer code to store, on said server, a plurality of message files;

computer code to create a plurality of message headers, each of which corresponds to one of said plurality of message files;

computer code to store, on said server, a document file having visually perceivable information associated therewith, with a subportion of said visually perceivable information being defined by a plurality of view parameters, with a subset of said plurality of message files having differing view parameters associated therewith; and

computer code to display a subgroup of said plurality of message headers on said client terminal, each of which recites a summary of content of one of the plurality of message files corresponding to the message headers of said subgroup.

2. The computer product as recited in claim 1 further including computer code to create, on said client terminal, a subgroup of said plurality of message files and subsequently store said subgroup on said server.

3. The computer product as recited in claim 1 wherein said client terminal includes memory and further including computer code to place said document file and one of said message files of said subset in said memory, with said viewable information associated with said document file being related to the view parameters associated with message file of said subset present in said memory.

4. The computer product as recited in claim 1 wherein said client terminal includes a processor and a memory and further including computer code to place said document file and multiple message files of said subset in said memory,

operating on one of said multiple message files of said subset with said processor, defining an executed file, with said viewable information associated with said document file being related to the view parameters associated with said executed message file.

5. The computer product as recited in claim 1 wherein said client terminal includes memory and said viewable information has annotation data associated therewith, further including computer code to place said document file and one of said message files of said subset in said memory, with said annotation data being independent of the view parameters associated with the message file of said subset present in said memory.

6. The computer product as recited in claim 1 further including computer code to display one of said plurality of message files on said client terminal, defining a selected message file.

7. The computer product as recited in claim 6 wherein said client terminal includes memory and said computer code to display one of said plurality of message files includes computer code to poll said memory to determine if said selected message file is present in said memory and display said selected message file if present therein and transmit a call to said server to send said selected message file upon determining said selected message file is not present in said memory.

8. The computer product as recited in claim 1 wherein said document file includes a file having a format selected from the set consisting of Drawing Web Format (DWF), a Autodesk AutoCAD format (DWG) ,a Drawing Exchange Format (DXF), a Simple Vector Format (SVF), a Bentley Microstation format, a Adobe Illustrator format (AI), a Corel Draw format (CMX), a Microsoft Powerpoint format (PPT), an HPGL, an HPCL/2 format, a Windows Bitmap file format (BMP), a Compuserve file format (GIF), a Tagged Image File Format (TIFF), a Joint Photographic Experts Group file format (JPEG), a Device

Independent Bitmap file format (DIB), a Targa file format (TGA), a PCX file format, an ASCII text file (TXT), a Rich Text Format file (RTF) and a Microsoft Word file (DOC), a Hypertext Markup Language file format (HTML), an Adobe Portable Document Format (PDF), a Windows Metafile (WMF), and a Postscript file format.

9. The computer product as recited in claim 1 wherein said message file includes an audio file embedded therein.

10. The computer product as recited in claim 1 wherein said message file includes a file having a format selected from the set consisting of Drawing Web Format (DWF), a Autodesk AutoCAD format (DWG) ,a Drawing Exchange Format (DXF), a Simple Vector Format (SVF), a Bentley Microstation format, a Adobe Illustrator format (AI), a Corel Draw format (CMX), a Microsoft Powerpoint format (PPT), an HPGL, an HPCL/2 format, a Windows Bitmap file format (BMP), a Compuserve file format (GIF), a Tagged Image File Format (TIFF), a Joint Photographic Experts Group file format (JPEG), a Device Independent Bitmap file format (DIB), a Targa file format (TGA), a PCX file format, an ASCII text file (TXT), a Rich Text Format file (RTF) and a Microsoft Word file (DOC), a Hypertext Markup Language file format (HTML), an Adobe Portable Document Format (PDF), a Windows Metafile (WMF), a Postscript file format, Motion Picture Experts Group file format (MPEG), Microsoft Video for Windows format (AVI), and Quicktime file format (MOV).

11. A computer product of the type comprising a computer readable medium that contains a program to control a network having a server and a client terminal having a display and a memory associated therewith, said computer product comprising:

computer code to create a searchable database, having message data, view parameter data and a document data, said document data having visually perceivable information associated therewith, with a subset of said message data corresponding to a subgroup of said view parameter data; and computer code to place said document file data and a subpart of said subset of said message data in said memory, with the viewable information associated with said document file being related to the view parameters associated with said subpart of said subset of said message data present in said memory.

12. The computer product as recited in claim 11 wherein said client terminal includes a processor and further including computer code to place said document file and multiple message files of said subset in said memory, operating on a portion of said subset of said message data with said processor, defining an executed file, with said viewable information associated being related to the view parameters associated said portion of said subset of said message data.

13. The computer product as recited in claim 12 further including computer code to store said searchable database on said server and displaying said portion of said subset of said message data on said client terminal.

14. The computer product as recited in claim 12 wherein said message data includes message content information and message header information, with said message header information providing a description of said message content information, and said computer code to place include computer code to transfer, in response to a request by said client terminal, said message header information from said server to said client terminal independent of transmitting said message content associated therewith independent of transmitting said message content information.

15. The computer product as recited in claim 11 wherein said computer code to create includes computer code to store said document data and said message data on said server, with said message data including a plurality of message files and message headers, with each of said plurality of message headers corresponding to one of said plurality of message files and said subset including said plurality of message files, said plurality of message files having differing view parameters associated therewith; and displaying a subgroup of said plurality of

message headers on said client terminal, each of which recites a summary of content of one of the plurality of message files corresponding to the message headers of said subgroup.

16. A computer product of the type comprising a computer readable medium that contains a program to control a network having a server and a client terminal having a display and a memory associated therewith, said computer product comprising:

computer code to store, on said server, a plurality of message files;

computer code to create a plurality of message headers, each of which corresponds to one of said plurality of message files;

computer code to store, on said server, a document file having visually perceivable information associated therewith, with a subportion of said visually perceivable information being defined by a plurality of view parameters, with a subset of said plurality of message files having differing view parameters associated therewith;

computer code to display a subgroup of said plurality of message headers on said client terminal, each of which recites a summary of content of one of the plurality of message files corresponding to the message headers of said subgroup; and

computer code to create, on said client terminal, a subgroup of said plurality of message files and subsequently store said subgroup on said server; and

computer code to place said document file and one of said message files of said subset in said memory, with said viewable information associated with said document file being related to the view parameters associated with message file of said subset present in said memory.

17. The computer product as recited in claim 16 wherein said client terminal includes a processor and further including computer code to place said document file and multiple message files of said subset in said memory, operating on one of said multiple message files of said subset with said processor, defining an

executed file, with said viewable information associated with said document file being related to the view parameters associated with said executed message file.

18. The computer product as recited in claim 17 wherein said viewable information has annotation data associated therewith, further including computer code to place said document file and one of said message files of said subset in said memory, with said annotation data being independent of the view parameters associated with the message file of said subset present in said memory.

19. The computer product as recited in claim 18 further including computer code to display one of said plurality of message files on said client terminal, defining a selected message file.

20. The computer product as recited in claim 19 wherein said computer code to display one of said plurality of message files includes computer code to poll said memory to determine if said selected message file is present in said memory and display said selected message file if present therein and transmit a call to said server to send said selected message file upon determining said selected message file is not present in said memory.

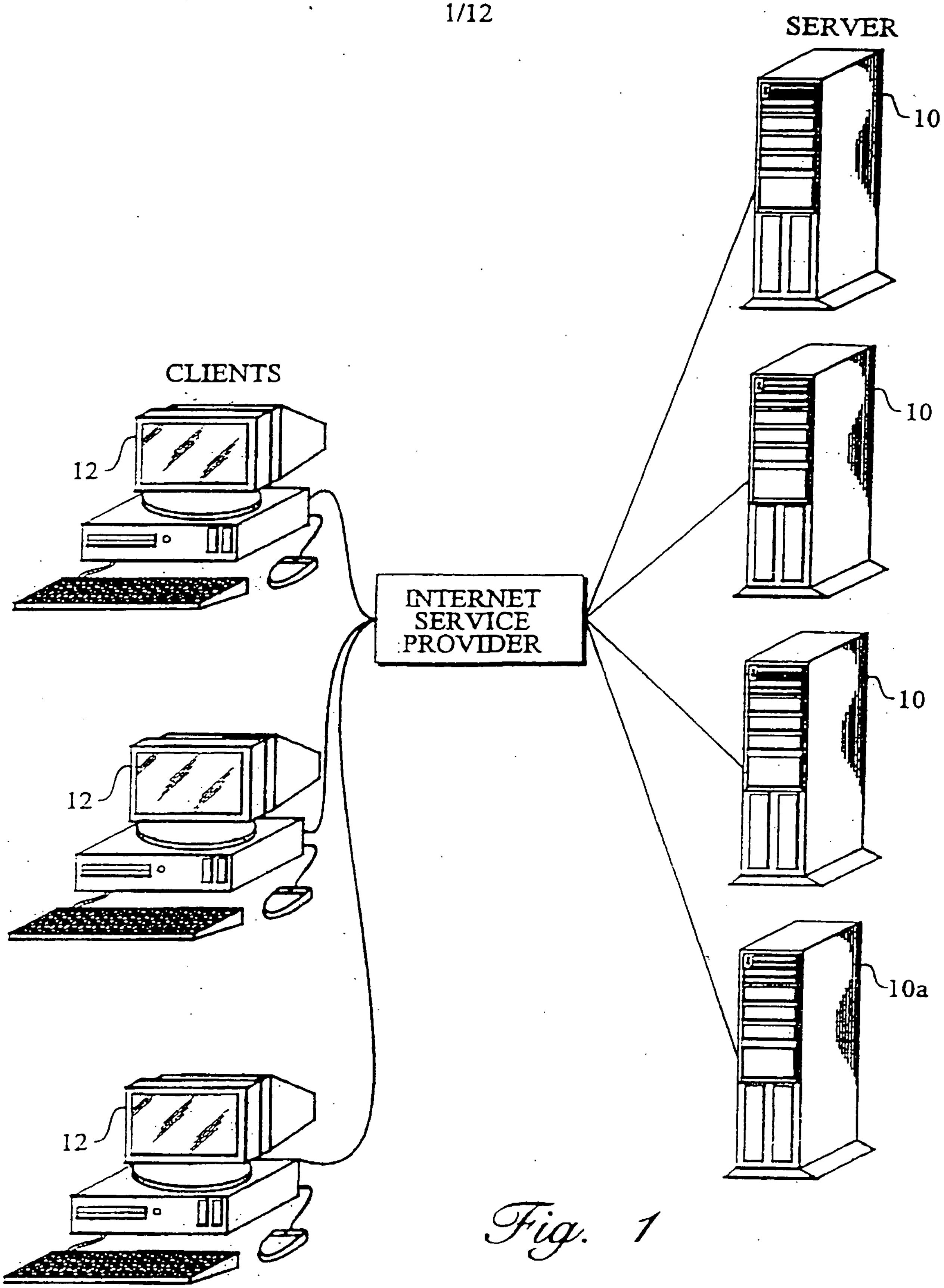

SERVER
10
10
10
10a
CLIENTS
12
12
12
INTERNET SERVICE PROVIDER

Fig. 1

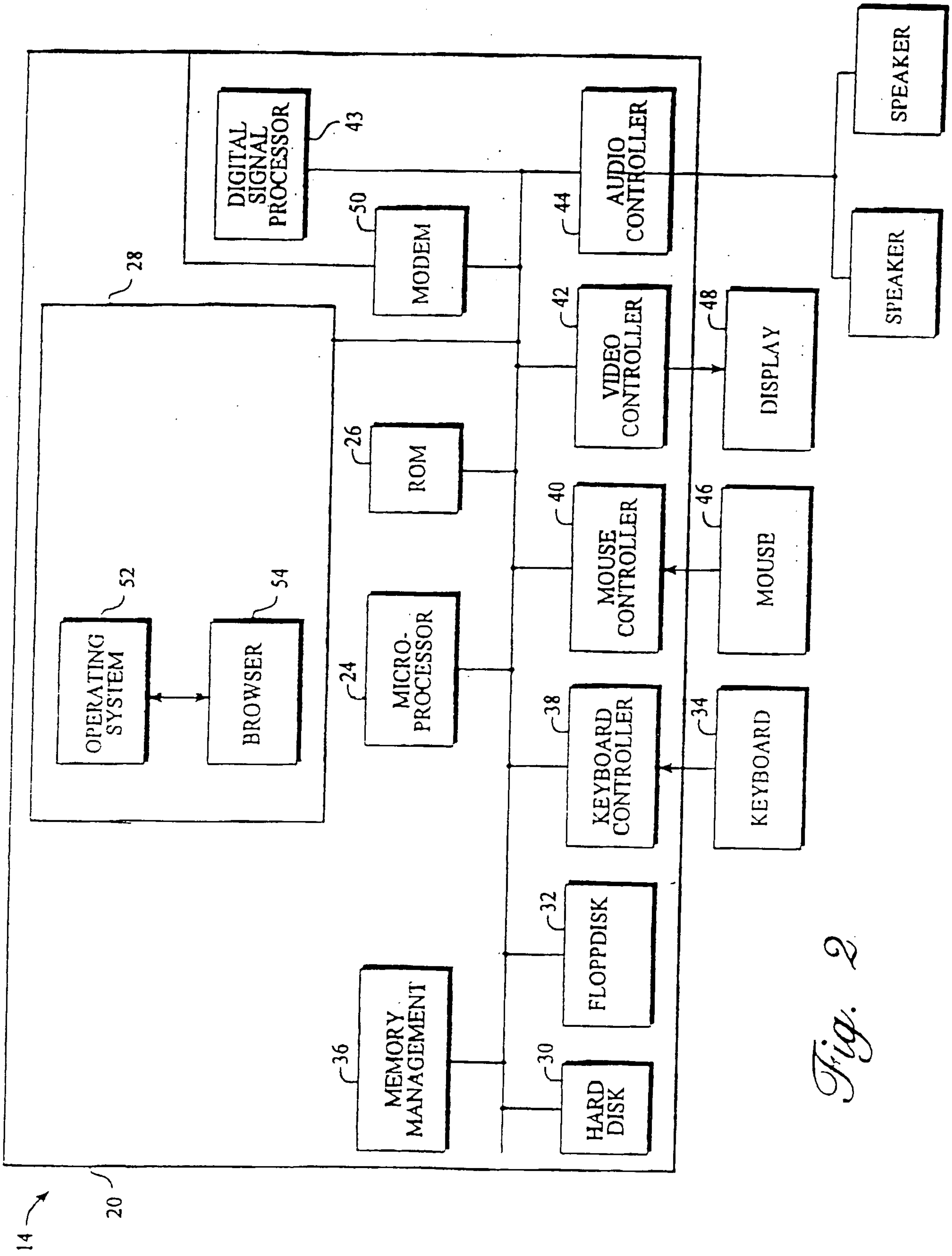

DIGITAL SIGNAL PROCESSOR
43
AUDIO CONTROLLER
44
SPEAKER
SPEAKER
MODEM
50
28
VIDEO CONTROLLER
42
DISPLAY
48
ROM
26
MOUSE CONTROLLER
40
MOUSE
46
OPERATING SYSTEM
52
BROWSER
54
MICRO-PROCESSOR
24
KEYBOARD CONTROLLER
38
KEYBOARD
34
FLOPPDISK
32
MEMORY MANAGEMENT
36
HARD DISK
30
20
14

Fig. 2

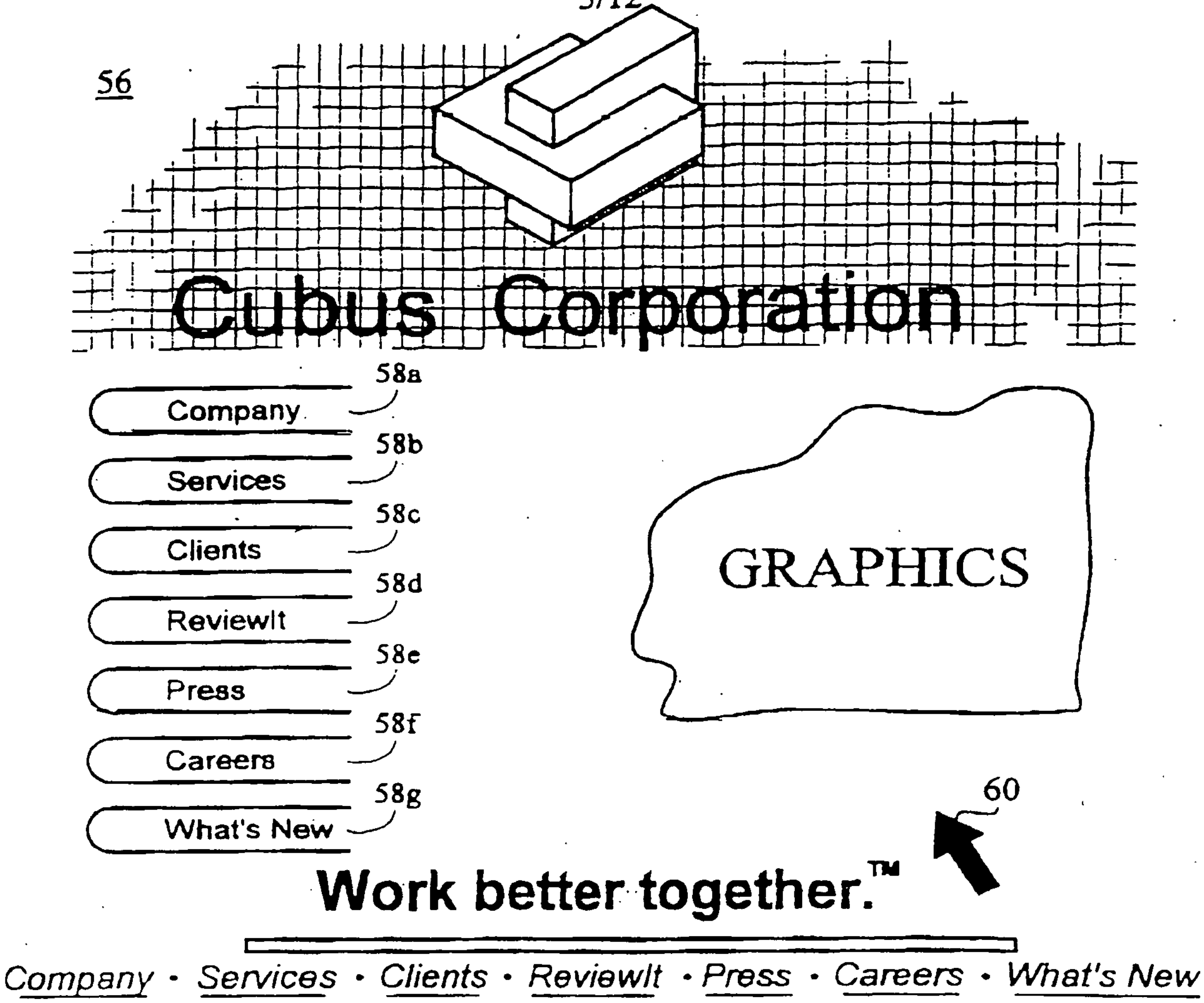
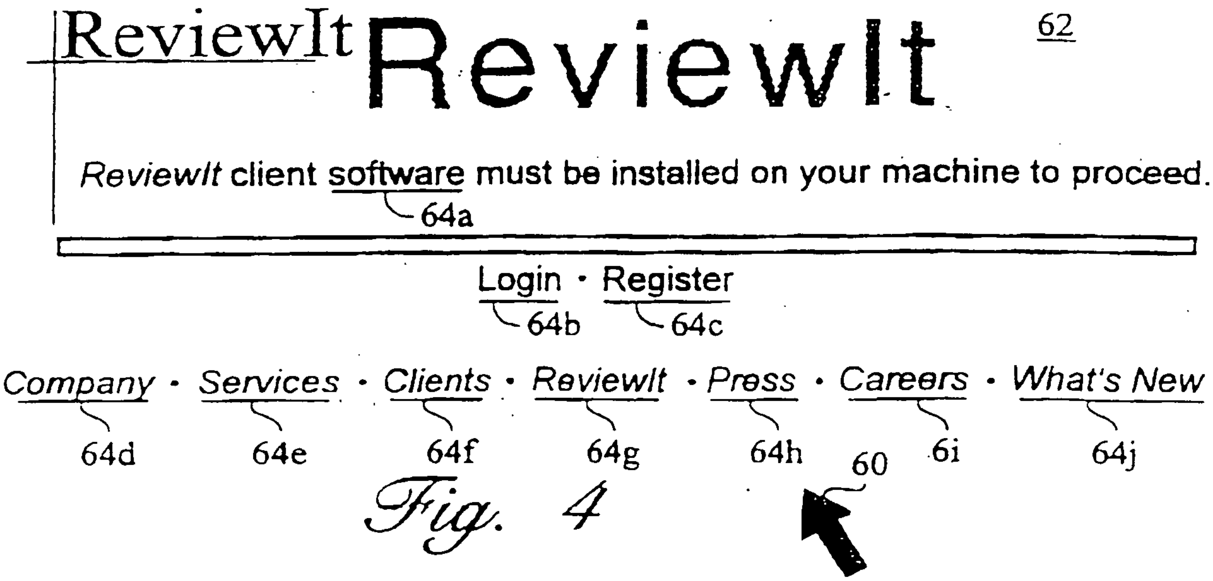
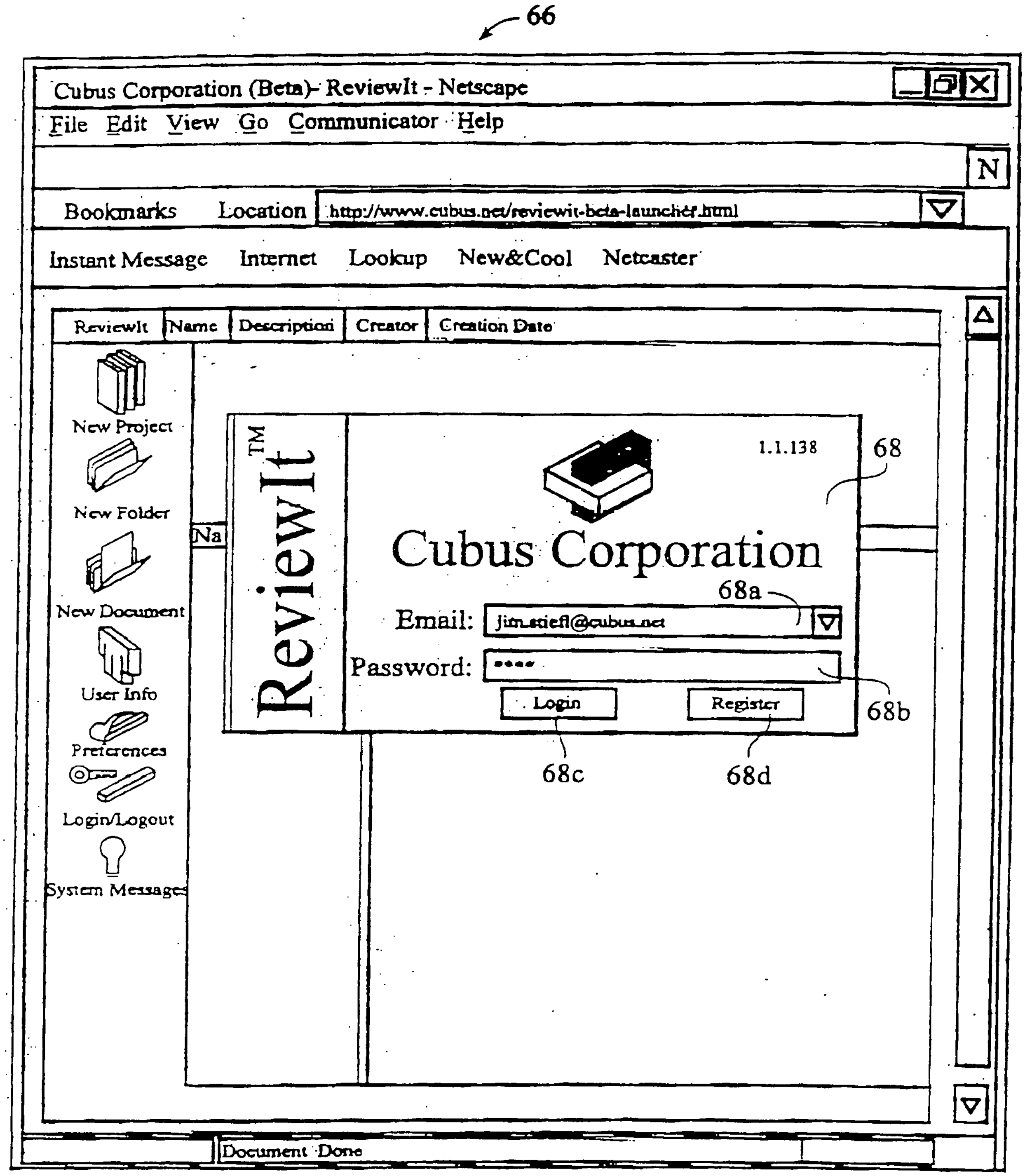


Fig. 3


ReviewIt
ReviewIt
62
64a
Login · Register
64b
64c
Company · Services · Clients · ReviewIt · Press · Careers · What's New
64d
64e
64f
64g
64h
60
6i
64j

ReviewIt client software must be installed on your machine to proceed.

Fig. 4



Location: http://www.cubus.net/reviewit-beta-launcher.html

1.1.138

Email: jim.stiefl@cubus.net

Password: ****

Document Done

Fig. 5

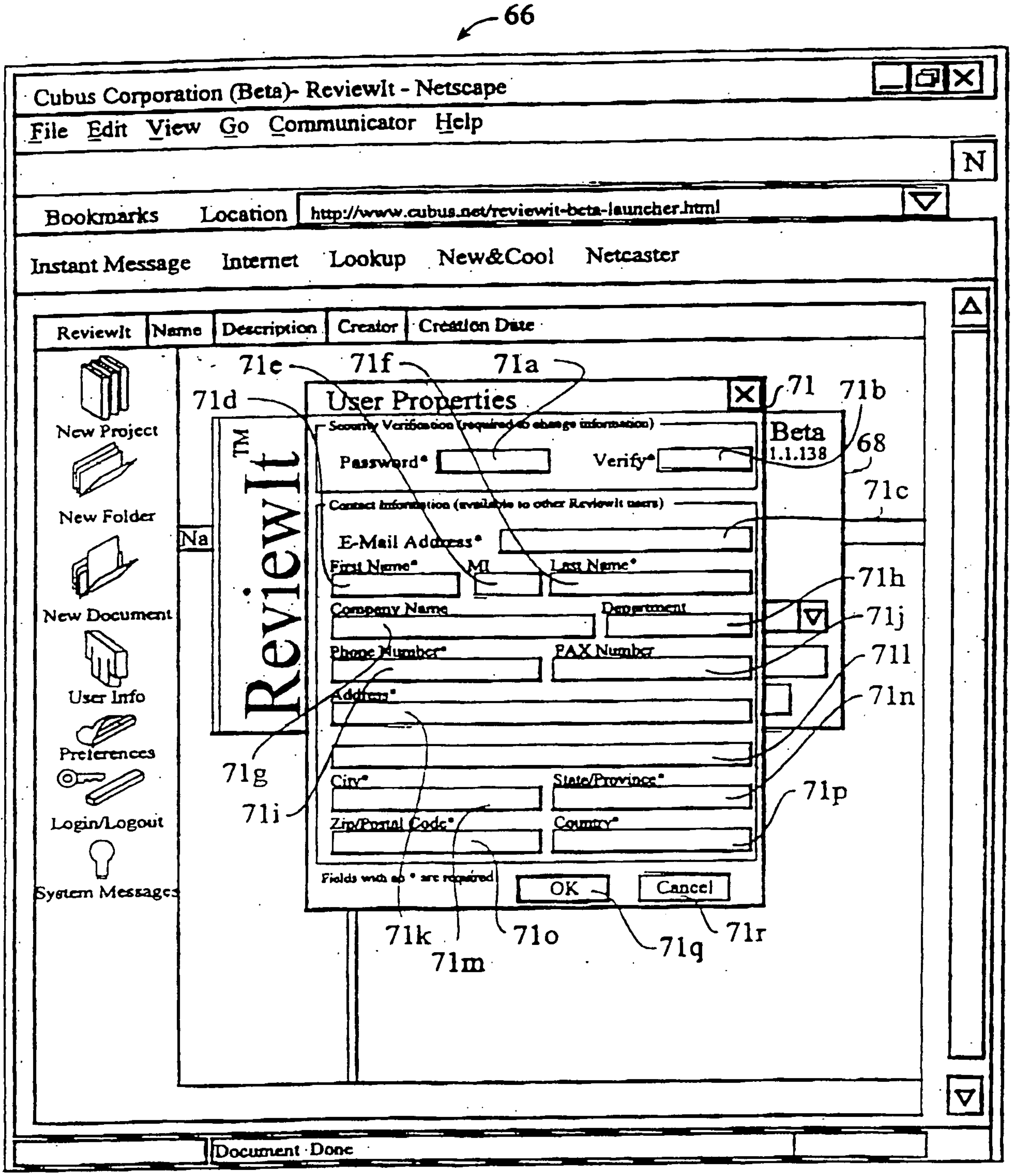
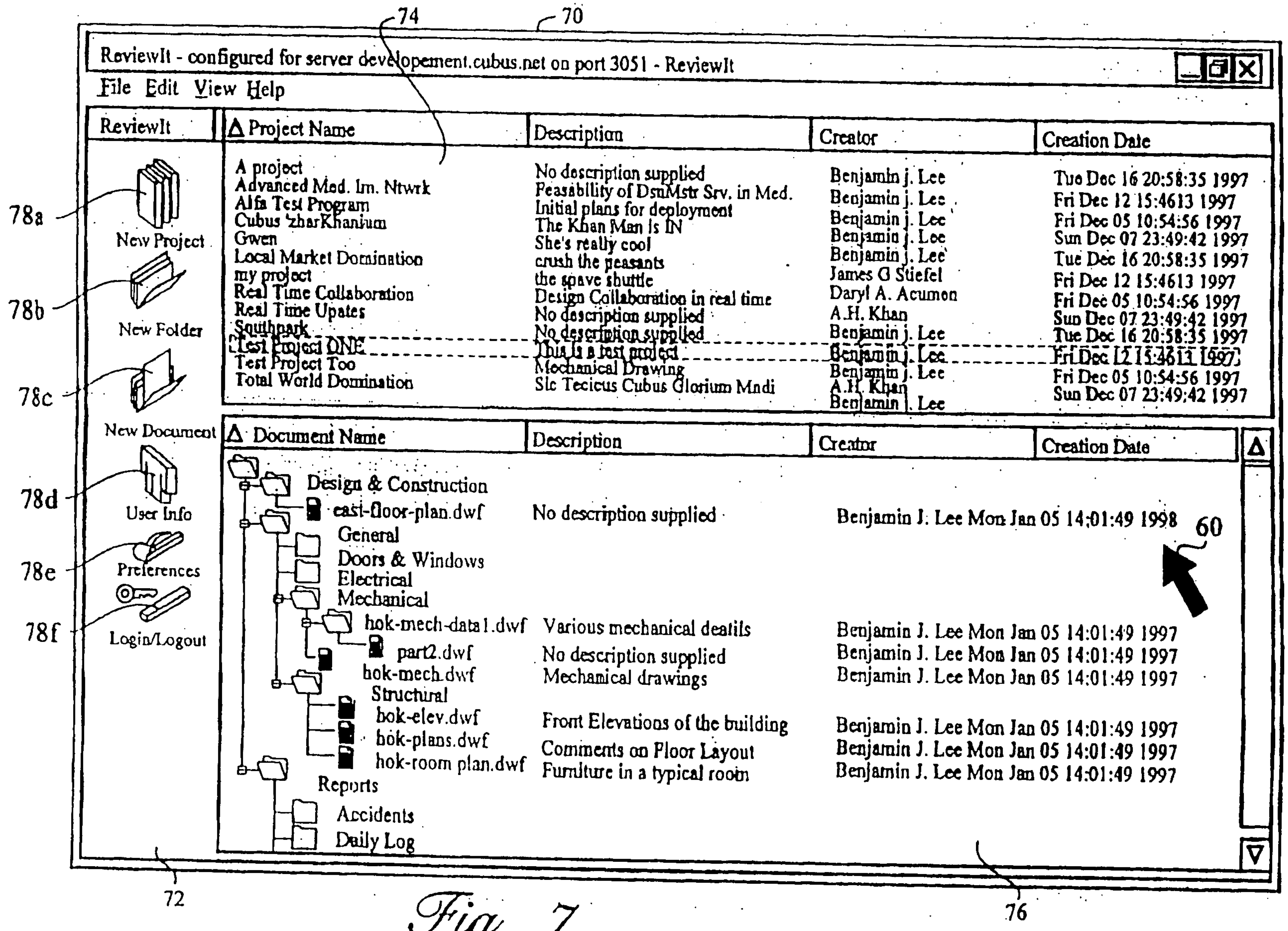

66
Cubus Corporation (Beta)- ReviewIt - Netscape
File Edit View Go Communicator Help
N
Bookmarks Location http://www.cubus.net/reviewit-beta-launcher.html
Instant Message Internet Lookup New&Cool Netcaster
ReviewIt Name Description Creator Creation Date
New Project
New Folder
New Document
User Info
Preferences
Login/Logout
System Messages
Na
ReviewIt™
Beta 1.1.138
User Properties
Security Verification (required to change information)
Password*
Verify*
Contact Information (available to other ReviewIt users)
E-Mail Address*
First Name*
MI
Last Name*
Company Name
Department
Phone Number*
FAX Number
Address*
City*
State/Province*
Zip/Postal Code*
Country*
Fields with an * are required
OK
Cancel
Document Done
71e
71f
71a
71d
71
71b
68
71c
71h
71j
71l
71n
71g
71i
71p
71k
71m
71o
71q
71r

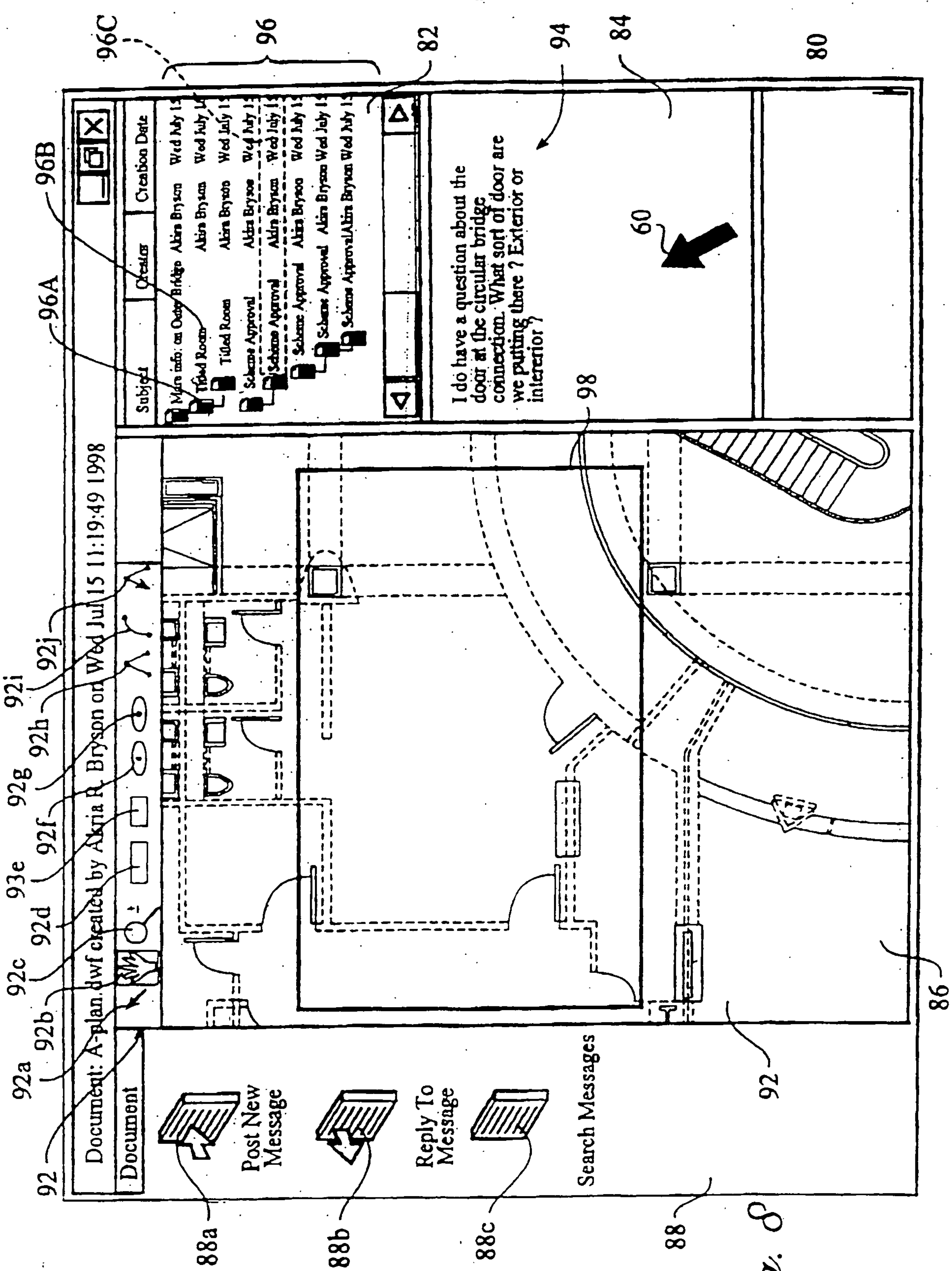
Fig. 6



Δ Project Name	Description	Creator	Creation Date
A project	No description supplied	Benjamin j. Lee	Tue Dec 16 20:58:35 1997
Advanced Med. Im. Ntwrk	Peasability of DsnMstr Srv. in Med.	Benjamin j. Lee	Fri Dec 12 15:4613 1997
Alfa Test Program	Initial plans for deployment	Benjamin j. Lee	Fri Dec 05 10:54:56 1997
Cubus 'zharKhanium	The Khan Man Is IN	Benjamin j. Lee	Sun Dec 07 23:49:42 1997
Gwen	She's really cool	Benjamin j. Lee	Tue Dec 16 20:58:35 1997
Local Market Domination	crush the peasants	James G Stiefel	Fri Dec 12 15:4613 1997
my project	the spave shuttle	Daryl A. Acumen	Fri Dec 05 10:54:56 1997
Real Time Collaboration	Design Collaboration in real time	A.H. Khan	Sun Dec 07 23:49:42 1997
Real Time Upates	No description supplied	Benjamin j. Lee	Tue Dec 16 20:58:35 1997
Southpark	No description supplied	Benjamin j. Lee	Fri Dec 12 15:4613 1997
Test Project ONE	This is a test project	Benjamin j. Lee	Fri Dec 05 10:54:56 1997
Test Project Too	Mechanical Drawing	A.H. Khan	Sun Dec 07 23:49:42 1997
Total World Domination	Slc Tecicus Cubus Glorium Mndi	Benjamin j. Lee	

Δ Document Name	Description	Creator	Creation Date
Design & Construction			
east-floor-plan.dwf	No description supplied	Benjamin J. Lee	Mon Jan 05 14:01:49 1998
General			
Doors & Windows			
Electrical			
Mechanical			
hok-mech-data1.dwf	Various mechanical deatils	Benjamin J. Lee	Mon Jan 05 14:01:49 1997
part2.dwf	No description supplied	Benjamin J. Lee	Mon Jan 05 14:01:49 1997
hok-mech.dwf	Mechanical drawings	Benjamin J. Lee	Mon Jan 05 14:01:49 1997
Structural			
hok-elev.dwf	Front Elevations of the building	Benjamin J. Lee	Mon Jan 05 14:01:49 1997
hok-plans.dwf	Comments on Floor Layout	Benjamin J. Lee	Mon Jan 05 14:01:49 1997
hok-room plan.dwf	Furniture in a typical room	Benjamin J. Lee	Mon Jan 05 14:01:49 1997
Reports			
Accidents			
Daily Log			

Fig. 7

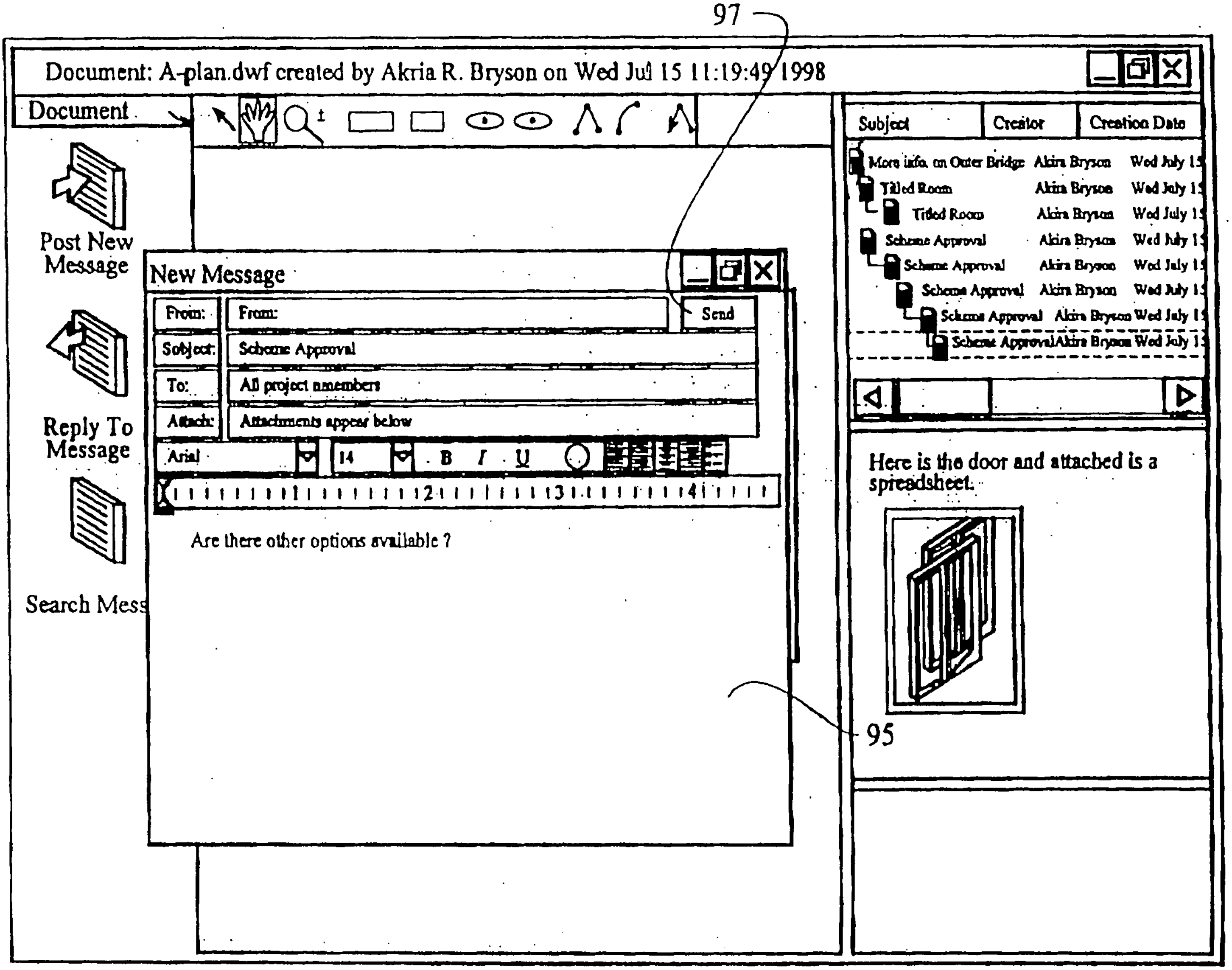


Document: A-plan.dwf created by Akria R. Bryson on Wed Jul 15 11:19:49 1998

Subject	Creator	Creation Date
More info. on Outer Bridge	Akira Bryson	Wed July 15
Tiled Room	Akira Bryson	Wed July 15
Tiled Room	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15

I do have a question about the door at the circular bridge connection. What sort of door are we putting there ? Exterior or interior ?

Fig. 8



Document: A-plan.dwf created by Akria R. Bryson on Wed Jul 15 11:19:49 1998

From: From:

Subject: Scheme Approval

To: All project nnembers

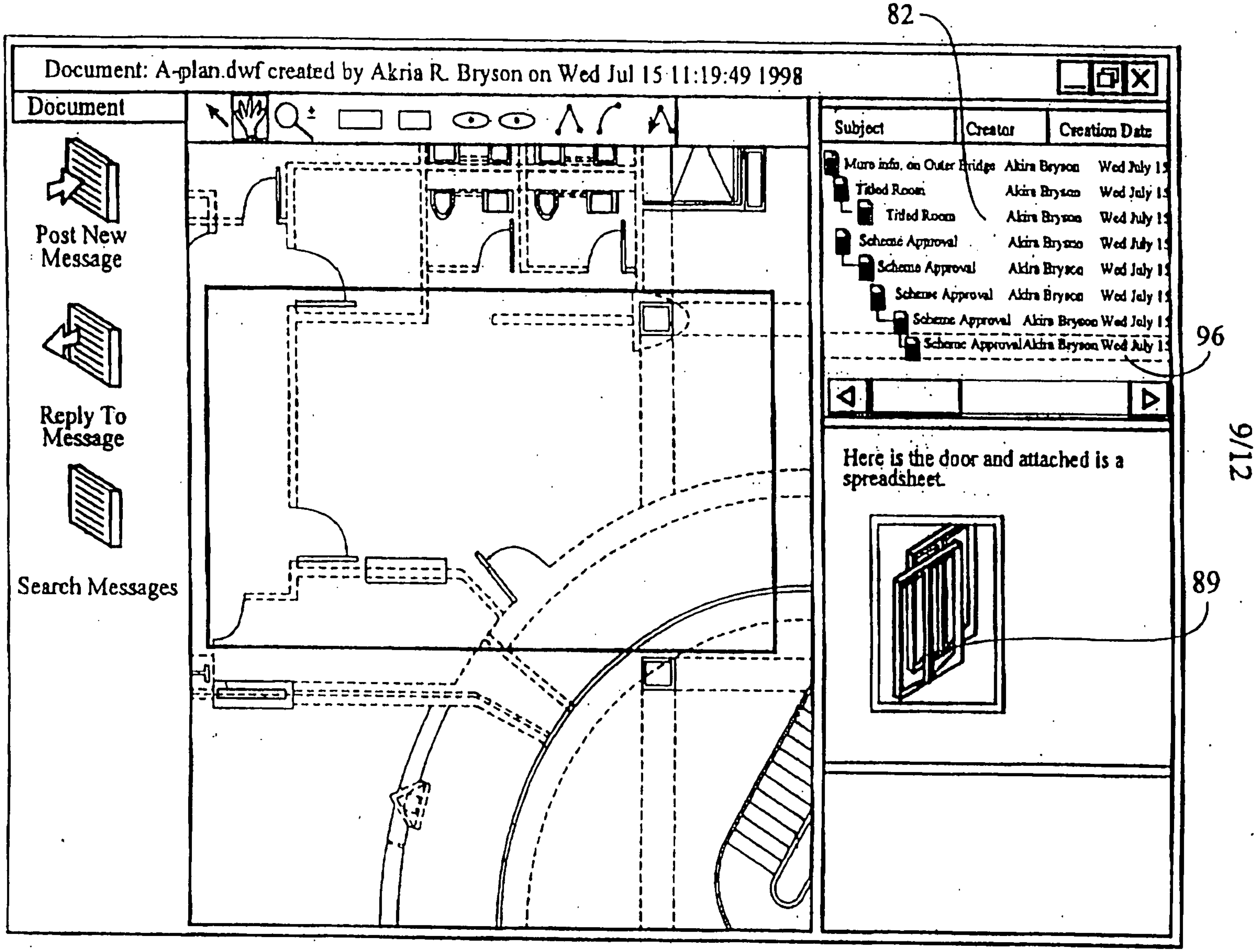
Attach: Attachments appear below

Are there other options available ?

Subject	Creator	Creation Date
More info. on Outer Bridge	Akira Bryson	Wed July 15
Tiled Room	Akira Bryson	Wed July 15
Titled Room	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15

Here is the door and attached is a spreadsheet.

Fig. 9



Document: A-plan.dwf created by Akria R. Bryson on Wed Jul 15 11:19:49 1998

Subject	Creator	Creation Date
More info. on Outer Bridge	Akira Bryson	Wed July 15
Titled Room	Akira Bryson	Wed July 15
Titled Room	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15
Scheme Approval	Akira Bryson	Wed July 15

Here is the door and attached is a spreadsheet.

Fig.10

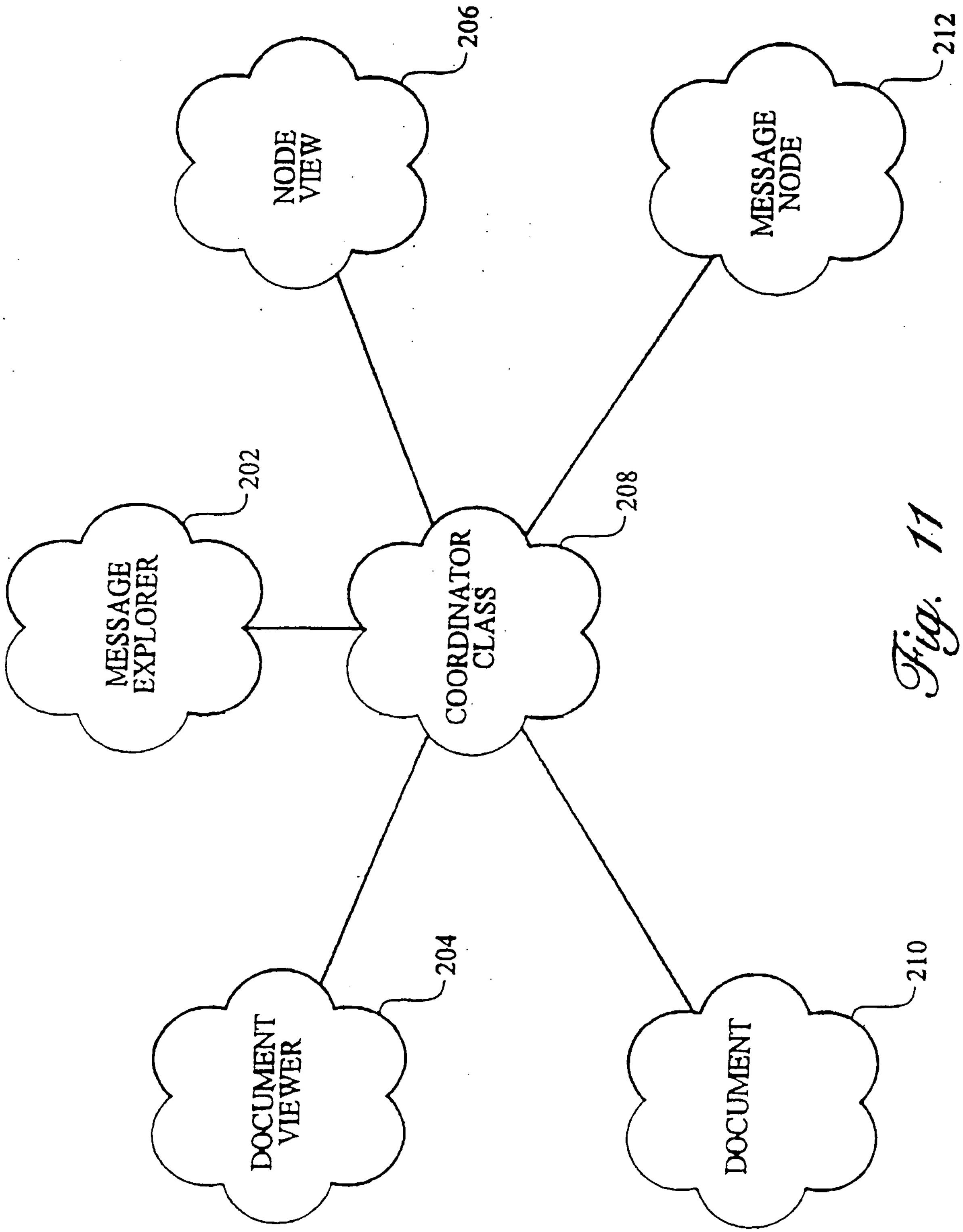

NODE VIEW
206
MESSAGE NODE
212
MESSAGE EXPLORER
202
COORDINATOR CLASS
208
DOCUMENT VIEWER
204
DOCUMENT
210

Fig. 11

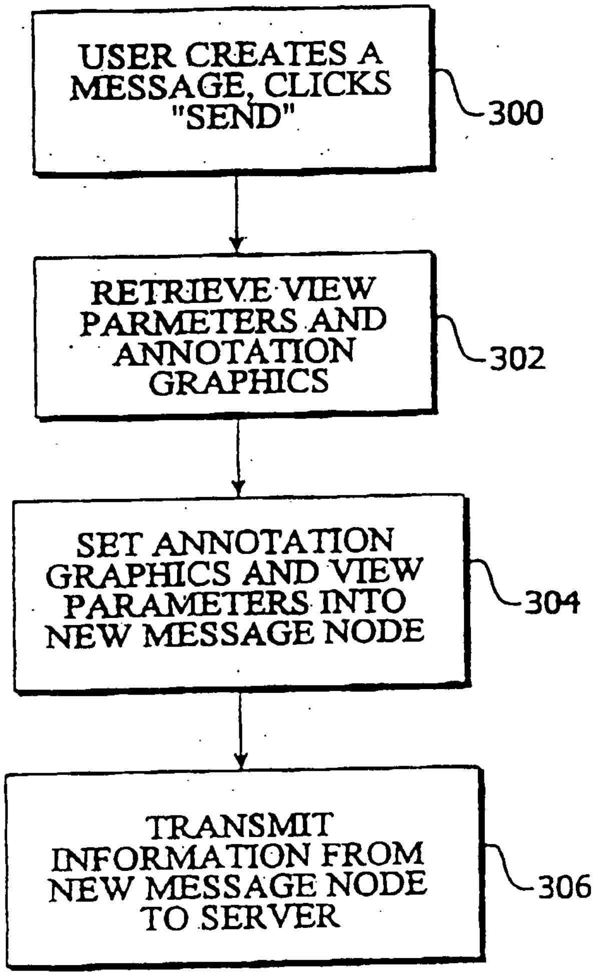

USER CREATES A MESSAGE, CLICKS "SEND"
300
RETRIEVE VIEW PARMETERS AND ANNOTATION GRAPHICS
302
SET ANNOTATION GRAPHICS AND VIEW PARAMETERS INTO NEW MESSAGE NODE
304
TRANSMIT INFORMATION FROM NEW MESSAGE NODE TO SERVER
306

Fig. 12

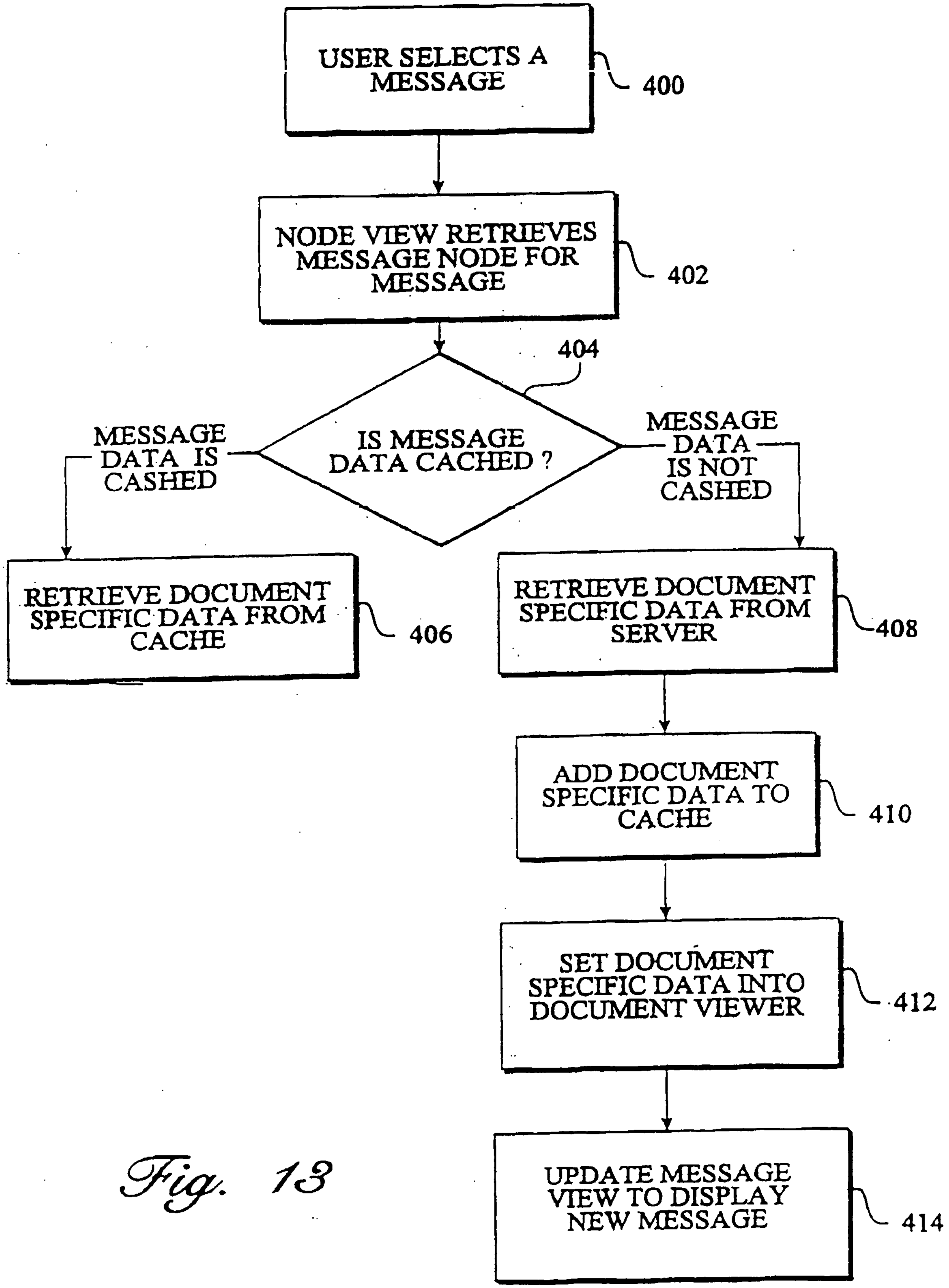

USER SELECTS A MESSAGE
400
NODE VIEW RETRIEVES MESSAGE NODE FOR MESSAGE
402
404
IS MESSAGE DATA CACHED ?
MESSAGE DATA IS CASHED
MESSAGE DATA IS NOT CASHED
RETRIEVE DOCUMENT SPECIFIC DATA FROM CACHE
406
RETRIEVE DOCUMENT SPECIFIC DATA FROM SERVER
408
ADD DOCUMENT SPECIFIC DATA TO CACHE
410
SET DOCUMENT SPECIFIC DATA INTO DOCUMENT VIEWER
412
UPDATE MESSAGE VIEW TO DISPLAY NEW MESSAGE
414

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/19074

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 15/16
US CL :709/203; 707/512
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/203; 707/512, 522, 526

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, WEST, DIALOG, NPL
search terms: message headers, position information

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,809,512 A (KATO) 15 September 1998	1-20
A	US 5,321,505 A (LEDDY) 14 June 1994	1-20
A	US 3,920,895 A (VIERI et al) 18 November 1975	1-20

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 06 DECEMBER 1999	Date of mailing of the international search report 22 DEC 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MICHAEL RAZA James R. Matthews Telephone No. (703) 305-3900

THIS PAGE LEFT BLANK